**Quiz Application**

**Aim:**

To create a quiz based application to reduce the manual time taken to compute.

**Abstract:**

The main objective of the project MCQ Quiz Application is to manage the details of students, examinations, marks, courses and papers. The project is totally at administrative end and thus only the administrator is granted the access. The purpose of the project is to build an

application to reduce the manual work for managing the MCQ quiz and we will follow to achieve

these objectives in this project.

- To create an appropriate platform for best managing of MCQ test;

- To overcome the time consuming issues and taking MCQ tests;

- To release the marks of the test taker as soon as possible;

- To manage the information of different tests.

**Problem Statement:**

The ‘MCQ Quiz Application’ project to be developed to overcome the time consuming problem of manual system. Apart from that in current system, checking the answer sheets after taking test, waste the examiners time, so this application will check the correct answer and save the examiner time and carry the examination in an effective manner. The users which are use this system don’t need to high computing knowledge and also system will inform them while entering invalid data. The aim of this project is to computerized the existing manual system and help the examiners to save their valuable time and important data. Apart from this, data which are exist in this system, will exist for long period of time and will be easy accessible. This project helps the examiners to manage their services in a good way and provide a better service to their users. The objective of this project is to manage the details of students, examinations, marks, courses and papers in a good manner. The performance of the application will be fully control by administrator and administrator can guaranty any one to access. The project will reduce the manual process in managing examinations and all issues regarding that. Functionalities of the project will be as following: - Able the examiners to punch the MCQ questions online; - Able the users to solve the questions online; - Examiners can manage the information regarding exam; - Correct answers will be evaluated by system (First it should be determining by examiner.

**Source Code:**

**Html code: (register.html)**

<!DOCTYPE html>  
<html>  
<head>  
<title>Login Form</title>  
<link rel="stylesheet" type="text/css" href="stylelogin.css">  
</head>  
<body >  
<div class="Login-form">  
<h1>Login form</h1>  
<form>  
<p>Fullname</p>  
<input type="text" name="firstname">  
<p>E-mail</p>  
<input type="email" name="email">  
<p>Password</p>  
<input type="password" name="passeord"><br><br>  
<a href="home.html" class="log">Login</a><br><br>  
If not a user:  
<a href="Index.html" class ="btn">register</a>  
</form>  
</div>  
</body>  
</html>

**Html code(Login.html)**

<!DOCTYPE html>  
<html>  
<head>  
<title>Login Form</title>  
<link rel="stylesheet" type="text/css" href="stylelogin.css">  
</head>  
<body >  
<div class="Login-form">  
<h1>Login form</h1>  
<form>  
<p>Fullname</p>  
<input type="text" name="firstname">  
<p>E-mail</p>  
<input type="email" name="email">  
<p>Password</p>  
<input type="password" name="passeord"><br><br>  
<a href="home.html" class="log">Login</a><br><br>  
If not a user:  
<a href="Index.html" class ="btn">register</a>  
</form>  
</div>  
</body>  
</html>

**Css Code: (register.css)**

\*{  
padding: 0;  
margin: 0;  
font-family: cursive;  
}  
body{  
height:100vh;  
background-size: cover;  
background-color: teal;  
background-image: url("bg.jpg") no repeat;  
}  
.Login-form{  
position: absolute;  
top: 40%;  
left: 50%;  
transform: translate(-50%,-50%);  
width: 350px;  
}  
.Login-form input[type=text],input[type=password],[type=email]{  
text-align: centre;  
font-size: 20px;  
text-transform: uppercase;  
margin: 2px 0;  
}  
.Login-form input[type=radio]{  
align-content: center;  
}  
h1{  
margin-left:68px  
}  
 p{  
font-size: 25px;  
margin: 15px 0;  
}  
input{  
border: 0;  
font-size: 16px;  
width:300px;  
}  
  
 button{  
top: 30%;  
left: 20%;  
text-align: center;  
width: 58px;  
height: 20px;  
margin-left: 115px;  
margin-top: 30px;  
}  
  
.btn{  
margin-left: 9px;  
font-size: 13px;  
padding: 0 3px;  
color: #000000;  
background-color: #ffffff;  
text-decoration: none;  
border: 1px solid;  
}  
.log{  
margin-left: 125px;  
font-size: 13px;  
padding: 0 3px;  
color: #000000;  
background-color: #ffffff;  
text-decoration: none;  
border: 1px solid;  
}

**Css Code: (login.css)**

**\*** {  
padding: 0;  
margin: 0;  
font-family: cursive;  
}  
body{  
height:100vh;  
background-size: cover;  
background-color: teal;  
background-image: url("bg.jpg") no repeat;  
}  
.Login-form{  
position: absolute;  
top: 40%;  
left: 50%;  
transform: translate(-50%,-50%);  
width: 350px;  
}  
.Login-form input[type=text],input[type=password],[type=email]{  
text-align: centre;  
font-size: 20px;  
text-transform: uppercase;  
margin: 2px 0;  
}  
.Login-form input[type=radio]{  
align-content: center;  
}  
h1{  
margin-left:68px  
}  
 p{  
font-size: 25px;  
margin: 15px 0;  
}  
input{  
border: 0;  
font-size: 16px;  
width:300px;  
}  
  
 button{  
top: 30%;  
left: 20%;  
text-align: center;  
width: 58px;  
height: 20px;  
margin-left: 115px;  
margin-top: 30px;  
}  
  
.btn{  
margin-left: 9px;  
font-size: 13px;  
padding: 0 3px;  
color: #000000;  
background-color: #ffffff;  
text-decoration: none;  
border: 1px solid;  
}  
.log{  
margin-left: 125px;  
font-size: 13px;  
padding: 0 3px;  
color: #000000;  
background-color: #ffffff;  
text-decoration: none;  
border: 1px solid;  
}

**Html code:(subject.html)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Quiz Application</title>

<link rel="icon" href="favicon.png">

<link rel="stylesheet" href="style.css">

<!-- FontAweome CDN Link for Icons-->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<!-- start Quiz button -->

<div class="start\_btn"><button>Select the subject !</button></div>

<div class="score\_text">

<!-- Here I've inserted Score Result from JavaScript -->

</div>

<div id="buttons">

<button id="ajp" ><a href="home.html">Advanced Java Programming</a></button>

<button class="daa">Design Analysis and Algorithm</button>

<button class="cd">Compiler Design</button>

</div>

</div>

<!-- Inside this JavaScript file I've inserted Questions and Options only -->

<script src="question.js"></script>

<!-- Inside this JavaScript file I've coded all Quiz Codes -->

<script src="script.js"></script>

</body>

**Css code(Subjects.css)**

\*{

padding: 0;

margin: 0;

font-family: cursive;

}

body{

height:100vh;

background-size: cover;

background-color: green;

background-image: url("bg.jpg") no repeat;

}

.Login-form{

position: absolute;

top: 40%;

left: 50%;

transform: translate(-50%,-50%);

width: 350px;

}

.Login-form input[type=text],input[type=password],[type=email]{

text-align: centre;

font-size: 20px;

text-transform: uppercase;

margin: 2px 0;

}

.Login-form input[type=radio]{

align-content: center;

}

h1{

margin-left:22px

}

p{

font-size: 25px;

margin: 15px 0;

}

input{

border: 0;

font-size: 16px;

width:300px;

}

button{

top: 30%;

left: 20%;

text-align: center;

width: 58px;

height: 20px;

margin-left: 115px;

margin-top: 30px;

}

.btn{

margin-top: 20px;

margin-left: 115px;

font-size: 13px;

padding: 0 3px;

color: #000000;

background-color: #ffffff;

text-decoration: none;

border: 1px solid;

}

**Html code:( home.html)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Quiz Application</title>

<link rel="icon" href="favicon.png">

<link rel="stylesheet" href="cssfilenew.css">

<!-- FontAweome CDN Link for Icons-->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<!-- start Quiz button -->

<div class="start\_btn"><button>Start Quiz</button></div>

<!-- Info Box -->

<div class="info\_box">

<div class="info-title"><span>Rules of this Quiz</span></div>

<div class="info-list">

<div class="info">1. You will have only <span>15 seconds</span> per each question.</div>

<div class="info">2. Once you select your answer, it can't be undone.</div>

<div class="info">3. You can't select any option once time goes off.</div>

<div class="info">4. You'll get points on the basis of your correct answers.</div>

</div>

<div class="buttons">

<button class="quit">Exit Quiz</button>

<button class="restart">Continue</button>

</div>

</div>

<!-- Quiz Box -->

<div class="quiz\_box">

<header>

<div class="title">Quiz Application</div>

<div class="timer">

<div class="time\_left\_txt">Remaining Time</div>

<div class="timer\_sec">15</div>

</div>

<div class="time\_line"></div>

</header>

<section>

<div class="que\_text">

<!-- Here I've inserted question from JavaScript -->

</div>

<div class="option\_list">

<!-- Here I've inserted options from JavaScript -->

</div>

</section>

<!-- footer of Quiz Box -->

<footer>

<div class="total\_que">

<!-- Here I've inserted Question Count Number from JavaScript -->

</div>

<button class="next\_btn">Next Question</button>

</footer>

</div>

<!-- Result Box -->

<div class="result\_box">

<div class="icon">

<i class="fas fa-crown"></i>

</div>

<div class="complete\_text">You've completed the Quiz!</div>

<div class="score\_text">

<!-- Here I've inserted Score Result from JavaScript -->

</div>

<div class="buttons">

<button class="restart">Replay Quiz</button>

<button class="quit">Quit Quiz</button>

</div>

</div>

<!-- Inside this JavaScript file I've inserted Questions and Options only -->

<script src="question.js"></script>

<!-- Inside this JavaScript file I've coded all Quiz Codes -->

<script src="script.js"></script>

</body>

</html>

**CSS Code: (cssfilenew.css)**

/\* importing google fonts \*/

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

body{

background: #00e600;

}

::selection{

color: #fff;

background: #33ff33;

}

.start\_btn,

.info\_box,

.quiz\_box,

.result\_box{

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),

0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.info\_box.activeInfo,

.quiz\_box.activeQuiz,

.result\_box.activeResult{

opacity: 1;

z-index: 5;

pointer-events: auto;

transform: translate(-50%, -50%) scale(1);

}

.start\_btn button{

font-size: 25px;

font-weight: 500;

color: #33ff33;

padding: 15px 30px;

outline: none;

border: none;

border-radius: 5px;

background: #fff;

cursor: pointer;

}

.info\_box{

width: 540px;

background: #fff;

border-radius: 5px;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.info\_box .info-title{

height: 60px;

width: 100%;

border-bottom: 1px solid lightgrey;

display: flex;

align-items: center;

padding: 0 30px;

border-radius: 5px 5px 0 0;

font-size: 20px;

font-weight: 600;

}

.info\_box .info-list{

padding: 15px 30px;

}

.info\_box .info-list .info{

margin: 5px 0;

font-size: 17px;

}

.info\_box .info-list .info span{

font-weight: 600;

color: #33ff33;

}

.info\_box .buttons{

height: 60px;

display: flex;

align-items: center;

justify-content: flex-end;

padding: 0 30px;

border-top: 1px solid lightgrey;

}

.info\_box .buttons button{

margin: 0 5px;

height: 40px;

width: 100px;

font-size: 16px;

font-weight: 500;

cursor: pointer;

border: none;

outline: none;

border-radius: 5px;

border: 1px solid #33ff33;

transition: all 0.3s ease;

}

.quiz\_box{

width: 550px;

background: #fff;

border-radius: 5px;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.quiz\_box header{

position: relative;

z-index: 2;

height: 70px;

padding: 0 30px;

background: #fff;

border-radius: 5px 5px 0 0;

display: flex;

align-items: center;

justify-content: space-between;

box-shadow: 0px 3px 5px 1px rgba(0,0,0,0.1);

}

.quiz\_box header .title{

font-size: 20px;

font-weight: 600;

}

.quiz\_box header .timer{

color: #000000;

background: #b3ffb3;

border: 1px solid #80ff80;

height: 45px;

padding: 0 8px;

border-radius: 5px;

display: flex;

align-items: center;

justify-content: space-between;

width: 145px;

}

.quiz\_box header .timer .time\_left\_txt{

font-weight: 400;

font-size: 17px;

user-select: none;

}

.quiz\_box header .timer .timer\_sec{

font-size: 18px;

font-weight: 500;

height: 30px;

width: 45px;

color: #fff;

border-radius: 5px;

line-height: 30px;

text-align: center;

background: #80ffcc;

border: 1px solid #00ff99;

user-select: none;

}

.quiz\_box header .time\_line{

position: absolute;

bottom: 0px;

left: 0px;

height: 3px;

background: #00ff99;

}

section{

padding: 25px 30px 20px 30px;

background: #fff;

}

section .que\_text{

font-size: 25px;

font-weight: 600;

}

section .option\_list{

padding: 20px 0px;

display: block;

}

section .option\_list .option{

background: #00e68a;

border: 1px solid #00ff99;

border-radius: 5px;

padding: 8px 15px;

font-size: 17px;

margin-bottom: 15px;

cursor: pointer;

transition: all 0.3s ease;

display: flex;

align-items: center;

justify-content: space-between;

}

section .option\_list .option:last-child{

margin-bottom: 0px;

}

section .option\_list .option:hover{

color: #009900;

background: #99ff99;

border: 1px solid #008000;

}

section .option\_list .option.correct{

color: #155724;

background: #d4edda;

border: 1px solid #c3e6cb;

}

section .option\_list .option.incorrect{

color: #721c24;

background: #f8d7da;

border: 1px solid #f5c6cb;

}

section .option\_list .option.disabled{

pointer-events: none;

}

section .option\_list .option .icon{

height: 26px;

width: 26px;

border: 2px solid transparent;

border-radius: 50%;

text-align: center;

font-size: 13px;

pointer-events: none;

transition: all 0.3s ease;

line-height: 24px;

}

.option\_list .option .icon.tick{

color: #23903c;

border-color: #23903c;

background: #d4edda;

}

.option\_list .option .icon.cross{

color: #a42834;

background: #f8d7da;

border-color: #a42834;

}

footer{

height: 60px;

padding: 0 30px;

display: flex;

align-items: center;

justify-content: space-between;

border-top: 1px solid lightgrey;

}

footer .total\_que span{

display: flex;

user-select: none;

}

footer .total\_que span p{

font-weight: 500;

padding: 0 5px;

}

footer .total\_que span p:first-child{

padding-left: 0px;

}

footer button{

height: 40px;

padding: 0 13px;

font-size: 18px;

font-weight: 400;

cursor: pointer;

border: none;

outline: none;

color: #fff;

border-radius: 5px;

background: #00cc00;

border: 1px solid #009933;

line-height: 10px;

opacity: 0;

pointer-events: none;

transform: scale(0.95);

transition: all 0.3s ease;

}

footer button:hover{

background: #00b33c;

}

footer button.show{

opacity: 1;

pointer-events: auto;

transform: scale(1);

}

.result\_box{

background: #fff;

border-radius: 5px;

display: flex;

padding: 25px 30px;

width: 450px;

align-items: center;

flex-direction: column;

justify-content: center;

transform: translate(-50%, -50%) scale(0.9);

opacity: 0;

pointer-events: none;

transition: all 0.3s ease;

}

.result\_box .icon{

font-size: 100px;

color: #00ff99;

margin-bottom: 10px;

}

.result\_box .complete\_text{

font-size: 20px;

font-weight: 500;

}

.result\_box .score\_text span{

display: flex;

margin: 10px 0;

font-size: 18px;

font-weight: 500;

}

.result\_box .score\_text span p{

padding: 0 4px;

font-weight: 600;

}

.result\_box .buttons{

display: flex;

margin: 20px 0;

}

.result\_box .buttons button{

margin: 0 10px;

height: 45px;

padding: 0 20px;

font-size: 18px;

font-weight: 500;

cursor: pointer;

border: none;

outline: none;

border-radius: 5px;

border: 1px solid #009933;

transition: all 0.3s ease;

}

.buttons button.restart{

color: #fff;

background: #1aff66;

}

.buttons button.restart:hover{

background: #00b33c;

}

.buttons button.quit{

color: #1aff66;

background: #fff;

}

.buttons button.quit:hover{

color: #fff;

background: #00b33c;

}

**JavaScript File: (script.js)**

//selecting all required elements

const start\_btn = document.querySelector(".start\_btn button");

const info\_box = document.querySelector(".info\_box");

const exit\_btn = info\_box.querySelector(".buttons .quit");

const continue\_btn = info\_box.querySelector(".buttons .restart");

const quiz\_box = document.querySelector(".quiz\_box");

const result\_box = document.querySelector(".result\_box");

const option\_list = document.querySelector(".option\_list");

const time\_line = document.querySelector("header .time\_line");

const timeText = document.querySelector(".timer .time\_left\_txt");

const timeCount = document.querySelector(".timer .timer\_sec");

// if startQuiz button clicked

start\_btn.onclick = ()=>{

info\_box.classList.add("activeInfo"); //show info box

}

// if exitQuiz button clicked

exit\_btn.onclick = ()=>{

info\_box.classList.remove("activeInfo"); //hide info box

}

// if continueQuiz button clicked

continue\_btn.onclick = ()=>{

info\_box.classList.remove("activeInfo"); //hide info box

quiz\_box.classList.add("activeQuiz"); //show quiz box

showQuetions(0); //calling showQestions function

queCounter(1); //passing 1 parameter to queCounter

startTimer(15); //calling startTimer function

startTimerLine(0); //calling startTimerLine function

}

let timeValue = 15;

let que\_count = 0;

let que\_numb = 1;

let userScore = 0;

let counter;

let counterLine;

let widthValue = 0;

const restart\_quiz = result\_box.querySelector(".buttons .restart");

const quit\_quiz = result\_box.querySelector(".buttons .quit");

// if restartQuiz button clicked

restart\_quiz.onclick = ()=>{

quiz\_box.classList.add("activeQuiz"); //show quiz box

result\_box.classList.remove("activeResult"); //hide result box

timeValue = 15;

que\_count = 0;

que\_numb = 1;

userScore = 0;

widthValue = 0;

showQuetions(que\_count); //calling showQestions function

queCounter(que\_numb); //passing que\_numb value to queCounter

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

startTimer(timeValue); //calling startTimer function

startTimerLine(widthValue); //calling startTimerLine function

timeText.textContent = "Time Left"; //change the text of timeText to Time Left

next\_btn.classList.remove("show"); //hide the next button

}

// if quitQuiz button clicked

quit\_quiz.onclick = ()=>{

window.location.reload(); //reload the current window

}

const next\_btn = document.querySelector("footer .next\_btn");

const bottom\_ques\_counter = document.querySelector("footer .total\_que");

// if Next Que button clicked

next\_btn.onclick = ()=>{

if(que\_count < questions.length - 1){ //if question count is less than total question length

que\_count++; //increment the que\_count value

que\_numb++; //increment the que\_numb value

showQuetions(que\_count); //calling showQestions function

queCounter(que\_numb); //passing que\_numb value to queCounter

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

startTimer(timeValue); //calling startTimer function

startTimerLine(widthValue); //calling startTimerLine function

timeText.textContent = "Time Left"; //change the timeText to Time Left

next\_btn.classList.remove("show"); //hide the next button

}else{

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

showResult(); //calling showResult function

}

}

// getting questions and options from array

function showQuetions(index){

const que\_text = document.querySelector(".que\_text");

//creating a new span and div tag for question and option and passing the value using array index

let que\_tag = '<span>'+ questions[index].numb + ". " + questions[index].question +'</span>';

let option\_tag = '<div class="option"><span>'+ questions[index].options[0] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[1] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[2] +'</span></div>'

+ '<div class="option"><span>'+ questions[index].options[3] +'</span></div>';

que\_text.innerHTML = que\_tag; //adding new span tag inside que\_tag

option\_list.innerHTML = option\_tag; //adding new div tag inside option\_tag

const option = option\_list.querySelectorAll(".option");

// set onclick attribute to all available options

for(i=0; i < option.length; i++){

option[i].setAttribute("onclick", "optionSelected(this)");

}

}

// creating the new div tags which for icons

let tickIconTag = '<div class="icon tick"><i class="fas fa-check"></i></div>';

let crossIconTag = '<div class="icon cross"><i class="fas fa-times"></i></div>';

//if user clicked on option

function optionSelected(answer){

clearInterval(counter); //clear counter

clearInterval(counterLine); //clear counterLine

let userAns = answer.textContent; //getting user selected option

let correcAns = questions[que\_count].answer; //getting correct answer from array

const allOptions = option\_list.children.length; //getting all option items

if(userAns == correcAns){ //if user selected option is equal to array's correct answer

userScore += 1; //upgrading score value with 1

answer.classList.add("correct"); //adding green color to correct selected option

answer.insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to correct selected option

console.log("Correct Answer");

console.log("Your correct answers = " + userScore);

}else{

answer.classList.add("incorrect"); //adding red color to correct selected option

answer.insertAdjacentHTML("beforeend", crossIconTag); //adding cross icon to correct selected option

console.log("Wrong Answer");

for(i=0; i < allOptions; i++){

if(option\_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer

option\_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option

option\_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option

console.log("Auto selected correct answer.");

}

}

}

for(i=0; i < allOptions; i++){

option\_list.children[i].classList.add("disabled"); //once user select an option then disabled all options

}

next\_btn.classList.add("show"); //show the next button if user selected any option

}

function showResult(){

info\_box.classList.remove("activeInfo"); //hide info box

quiz\_box.classList.remove("activeQuiz"); //hide quiz box

result\_box.classList.add("activeResult"); //show result box

const scoreText = result\_box.querySelector(".score\_text");

if (userScore > 3){ // if user scored more than 3

//creating a new span tag and passing the user score number and total question number

let scoreTag = '<span>and congrats! 🎉, You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag; //adding new span tag inside score\_Text

}

else if(userScore > 1){ // if user scored more than 1

let scoreTag = '<span>and nice 😎, You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag;

}

else{ // if user scored less than 1

let scoreTag = '<span>and sorry 😐, You got only <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';

scoreText.innerHTML = scoreTag;

}

}

function startTimer(time){

counter = setInterval(timer, 1000);

function timer(){

timeCount.textContent = time; //changing the value of timeCount with time value

time--; //decrement the time value

if(time < 9){ //if timer is less than 9

let addZero = timeCount.textContent;

timeCount.textContent = "0" + addZero; //add a 0 before time value

}

if(time < 0){ //if timer is less than 0

clearInterval(counter); //clear counter

timeText.textContent = "Time Off"; //change the time text to time off

const allOptions = option\_list.children.length; //getting all option items

let correcAns = questions[que\_count].answer; //getting correct answer from array

for(i=0; i < allOptions; i++){

if(option\_list.children[i].textContent == correcAns){ //if there is an option which is matched to an array answer

option\_list.children[i].setAttribute("class", "option correct"); //adding green color to matched option

option\_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option

console.log("Time Off: Auto selected correct answer.");

}

}

for(i=0; i < allOptions; i++){

option\_list.children[i].classList.add("disabled"); //once user select an option then disabled all options

}

next\_btn.classList.add("show"); //show the next button if user selected any option

}

}

}

function startTimerLine(time){

counterLine = setInterval(timer, 29);

function timer(){

time += 1; //upgrading time value with 1

time\_line.style.width = time + "px"; //increasing width of time\_line with px by time value

if(time > 549){ //if time value is greater than 549

clearInterval(counterLine); //clear counterLine

}

}

}

function queCounter(index){

//creating a new span tag and passing the question number and total question

let totalQueCounTag = '<span><p>'+ index +'</p> of <p>'+ questions.length +'</p> Questions</span>';

bottom\_ques\_counter.innerHTML = totalQueCounTag; //adding new span tag inside bottom\_ques\_counter

}

**JavaScript File: (question.js)**

let questions = [

{

numb: 1,

question: "Mechanism by which a single piece of code can manipulate many different data types without explicitly having a separate entity for each data type is known as ?",

answer: "Generics",

options: [

"Generics",

"Encapsulation",

"Abstraction",

"Polymorphism"

]

},

{

numb: 2,

question: "In Javascript each frame in a frame set is represented as a?",

answer: "frame object",

options: [

"object",

"frame",

"frame object",

"frame set"

]

},

{

numb: 3,

question: "The purpose of a DTD is to define legal building blocks of which document?",

answer: "XML",

options: [

"HTML",

"XML",

"CSS",

"JS"

]

},

{

numb: 4,

question: "Which of the following is not a method to read a parameters from a form?",

answer: "getParameter(String pname)",

options: [

"getParameterValues(String pname)",

"getParameterNames()",

"getParameter(String pname)",

"getParameterValues()"

]

},

{

numb: 5,

question: "What does XML stand for?",

answer: "eXtensible Markup Language",

options: [

"eXtensible Markup Language",

"eXecutable Multiple Language",

"eXTra Multi-Program Language",

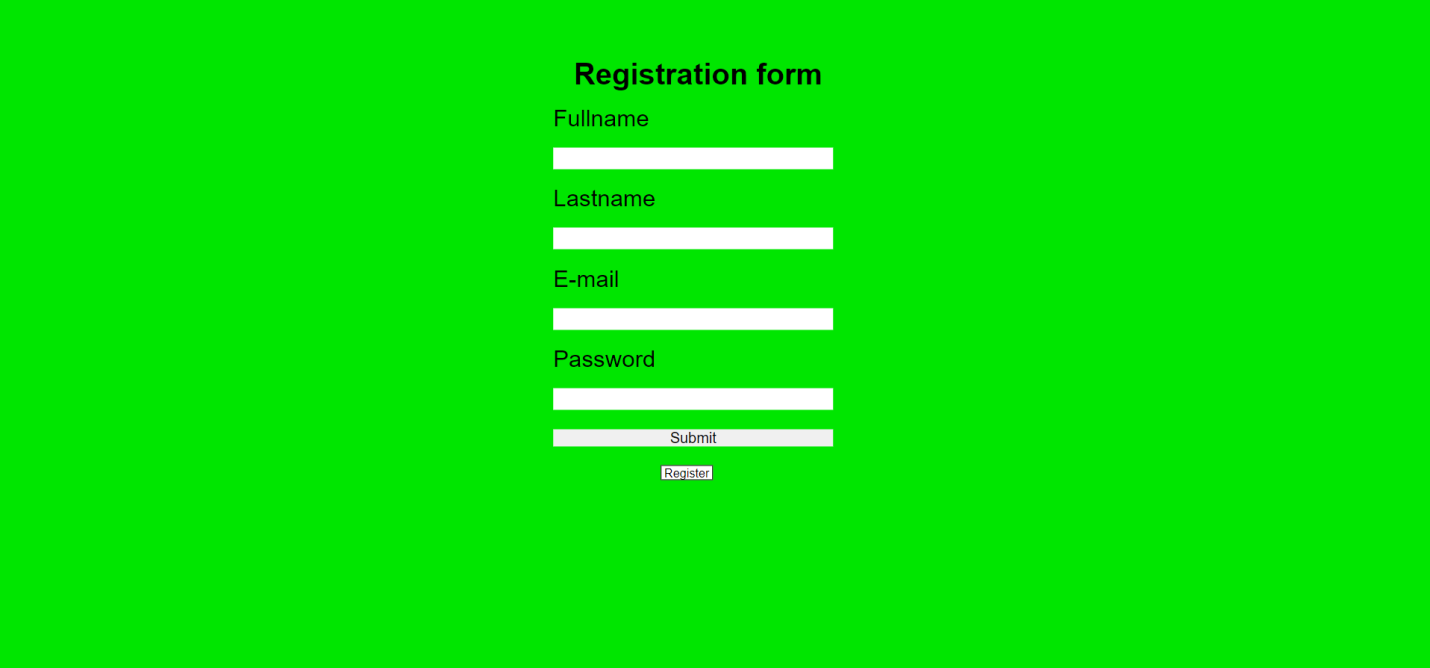
"eXamine Multiple Language"

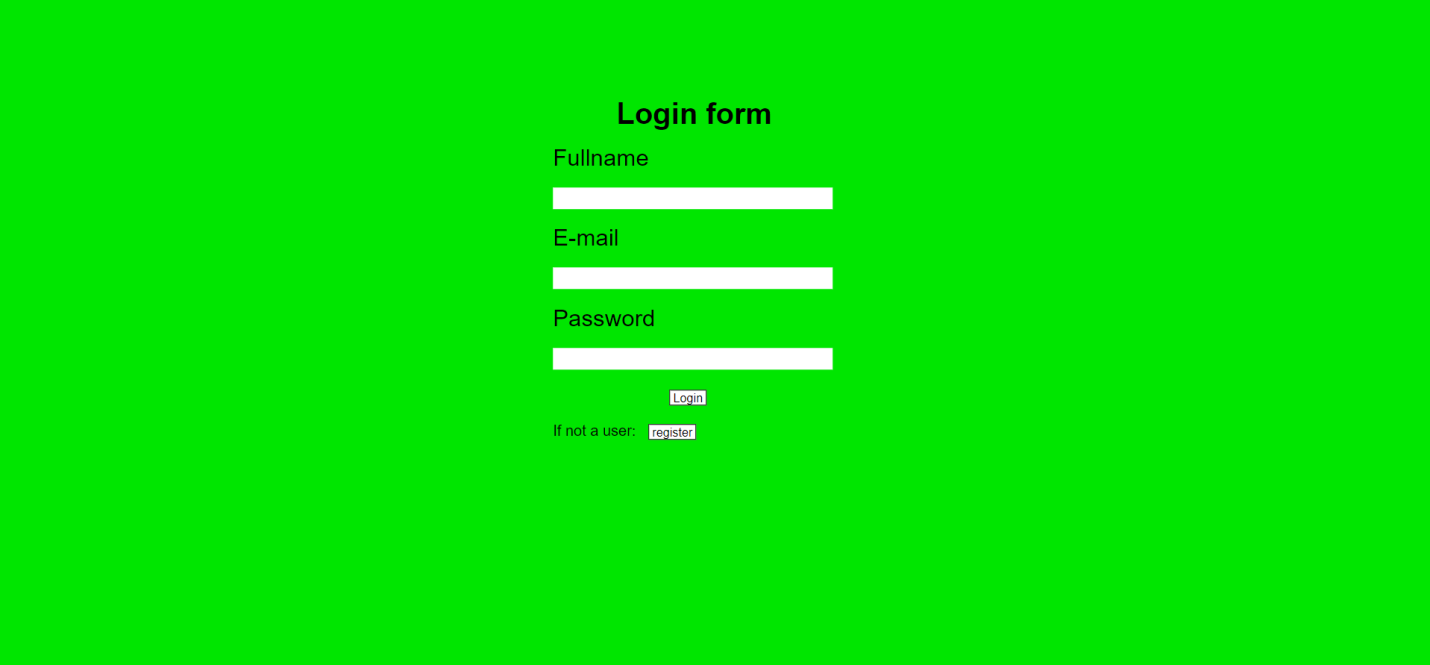
]

},

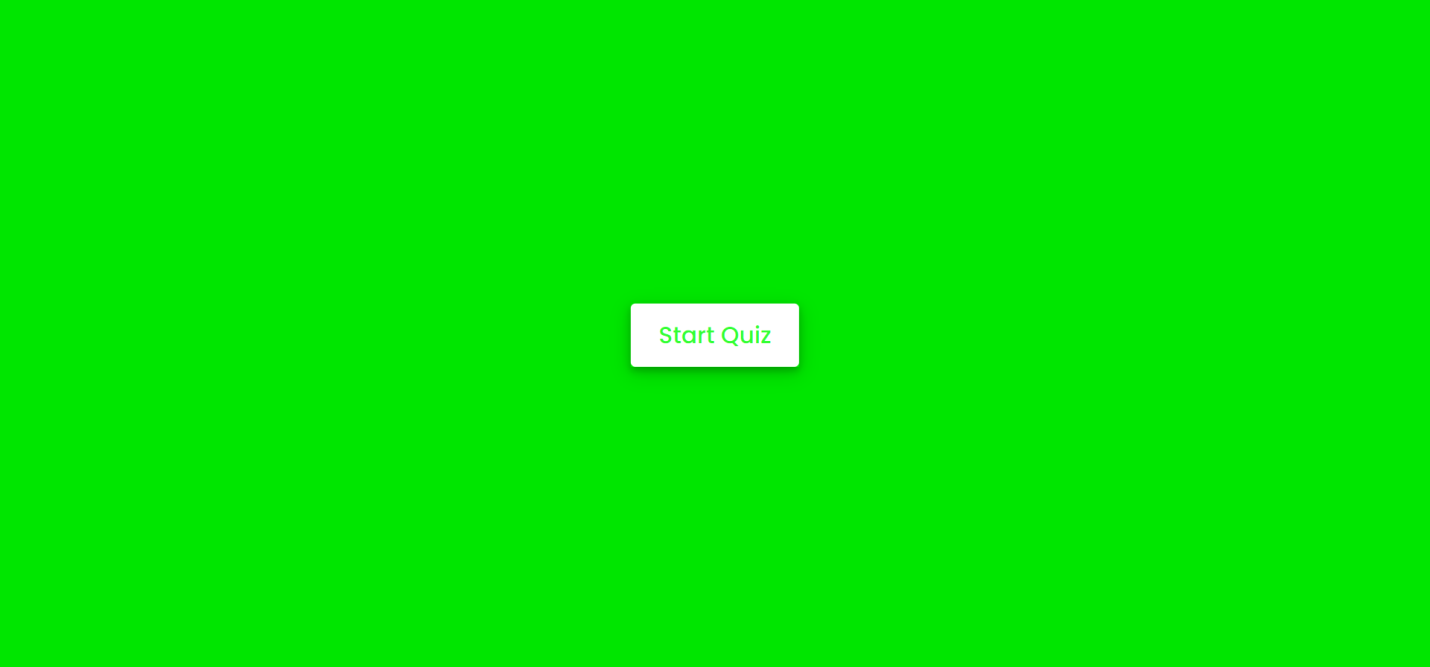
];

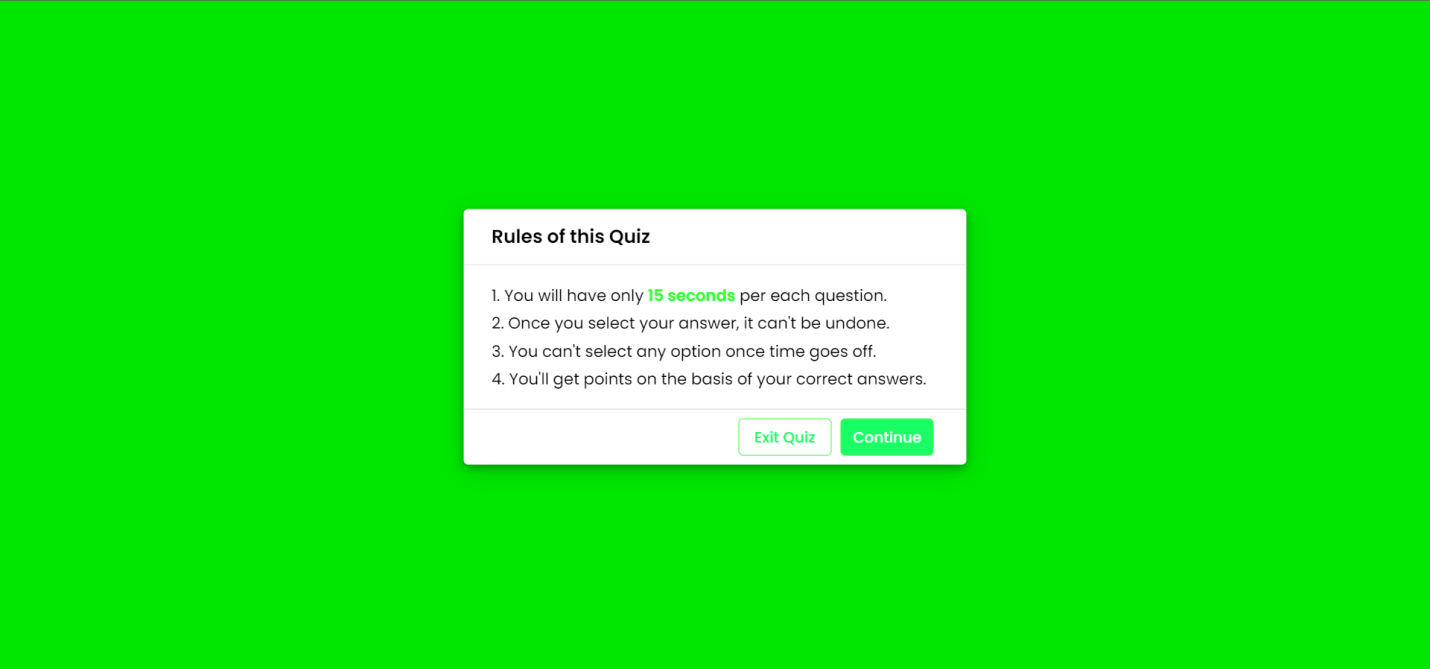
**Screenshots:**

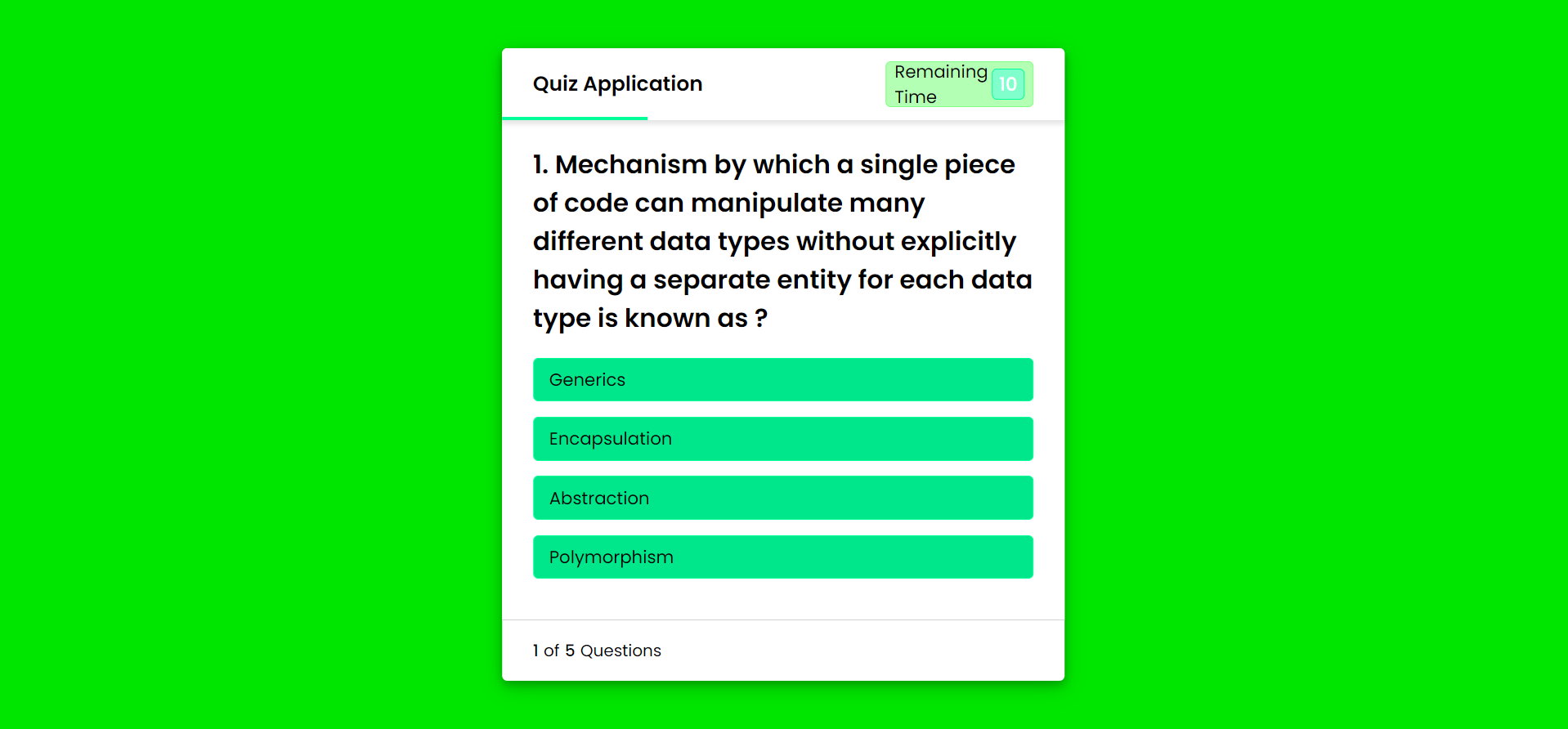
****

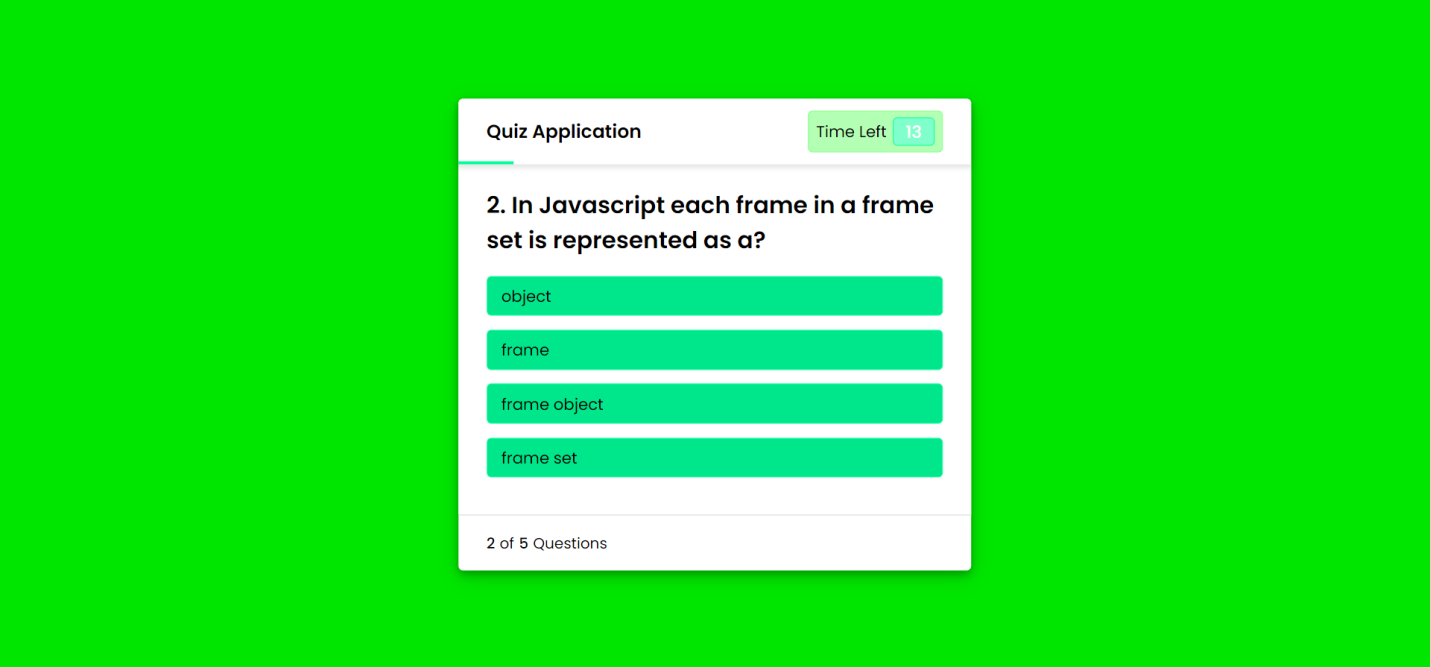


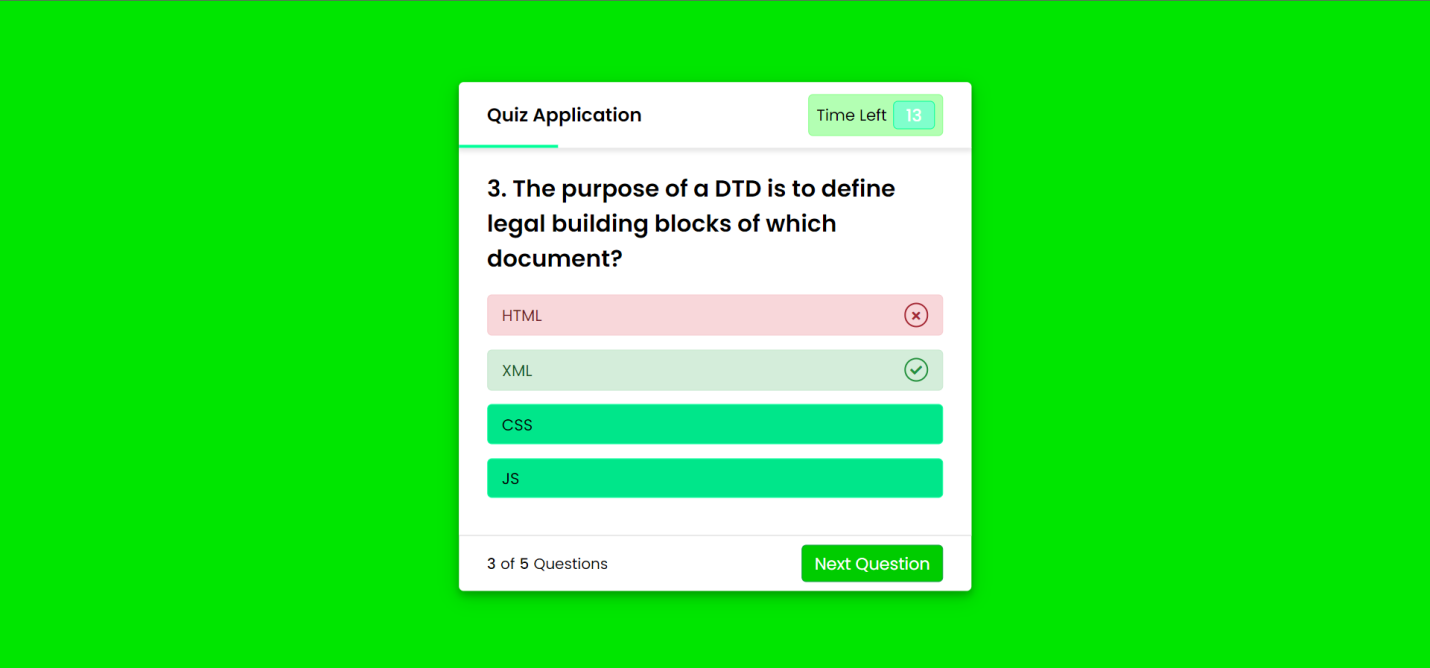


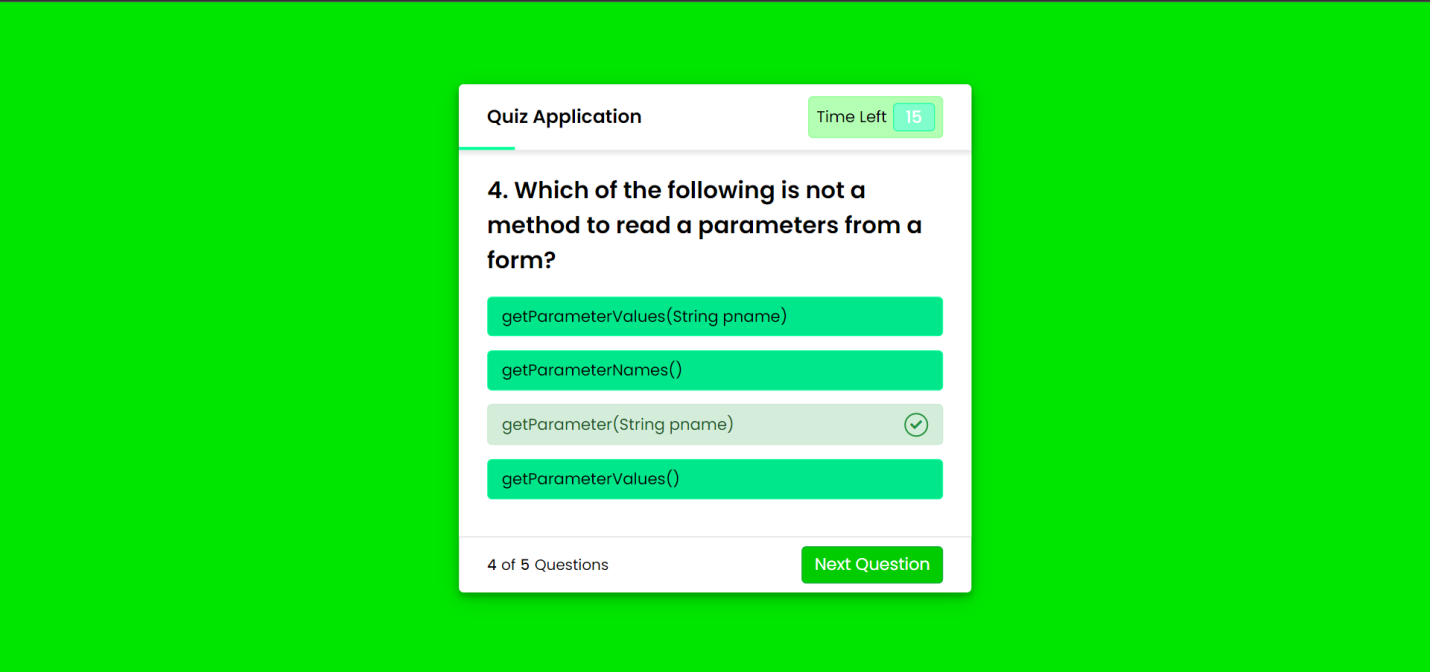


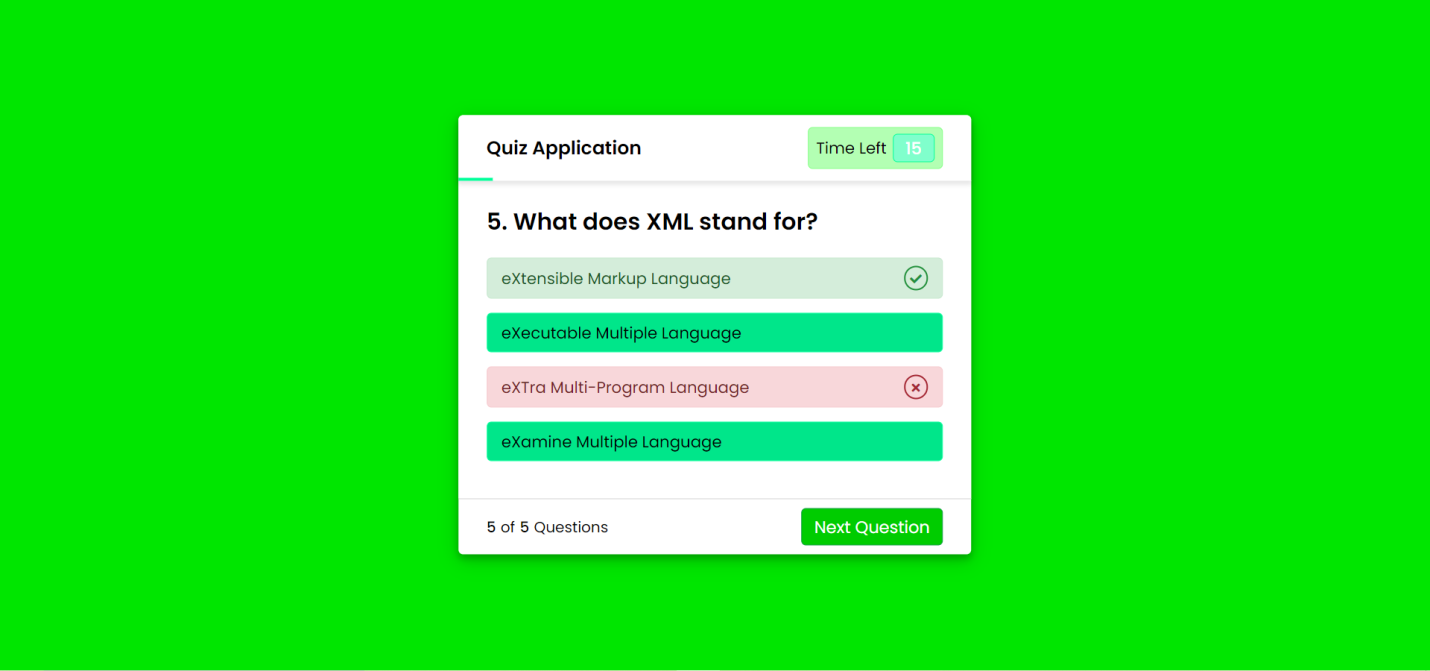


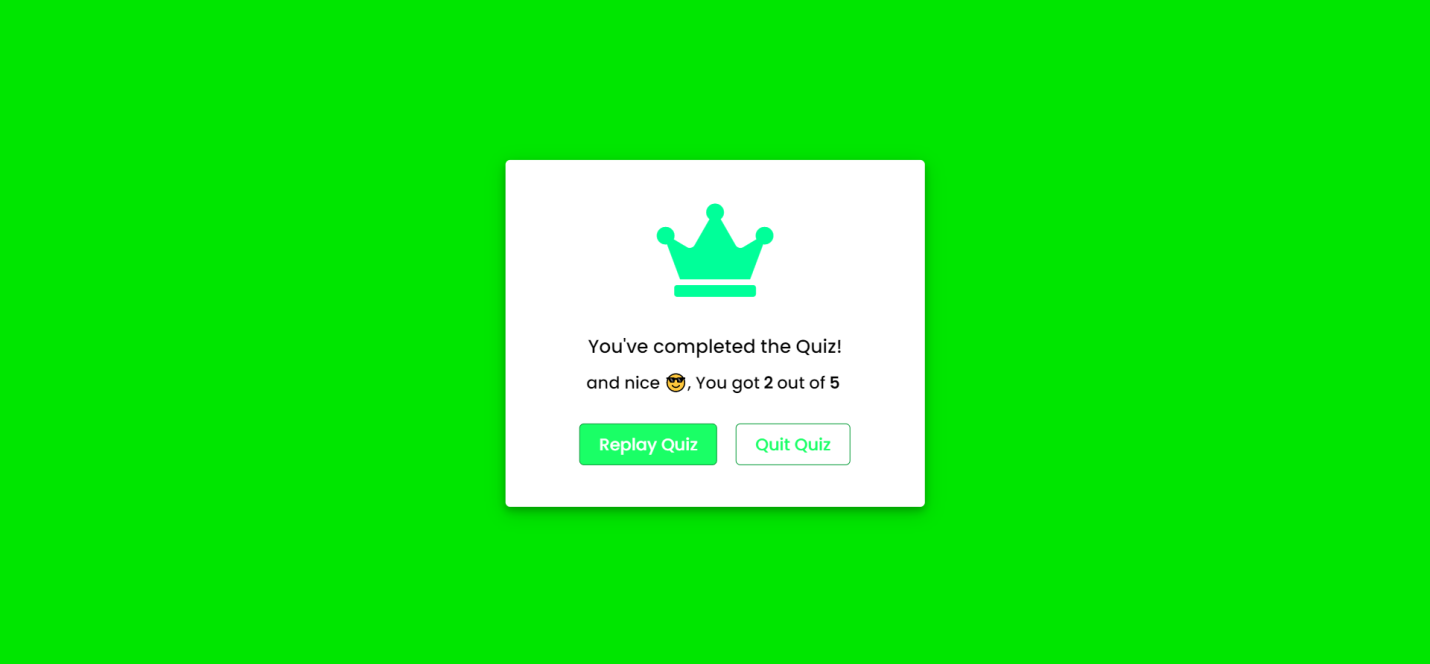












**Result:**

Hence we created a quiz based application to reduce the manual time taken to compute.